(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 848 545 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
17.06.1998 Bulletin 1998/25

(51) Int. Cl.^{6}: H04N 1/407

(21) Application number: 97203731.1

(22) Date of filing: 28.11.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 10.12.1996 US 763270

(71) Applicant: EASTMAN KODAK COMPANY
Rochester, New York 14650 (US)

(72) Inventors:
- Lee, Hsien-Che,
c/o Eastman Kodak Company
Rochester, New York 14650-2201 (US)
- Kwon, Heemin,
c/o Eastman Kodak Company
Rochester, New York 14650-2201 (US)

(74) Representative:
Lewandowsky, Klaus, Dipl.-Ing.
Kodak Aktiengesellschaft,
Patentabteilung
70323 Stuttgart (DE)

(54) Method for estimating and adjusting digital image contrast

(57) A method of estimating the scene contrast from a digital image, the method comprises the steps of: forming a Laplacian histogram distribution; determining, from the Laplacian histogram, first and second thresholds which eliminate substantially uniform areas or a substantially textured portion of the digital image; selecting pixels which are based on the first and second thresholds from the digital image; forming a histogram from the sampled pixels; computing a standard deviation of the sampled histogram; and estimating contrast of the digital image by comparing the computed standard deviation with a predetermined contrast for determining contrast of the input image in relationship with the predetermined contrast.

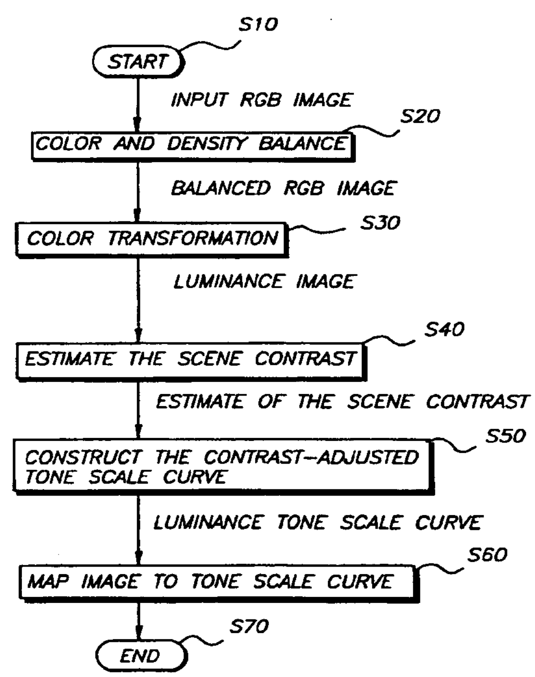

FIG. 2

Description

The present invention relates to digital imaging processing and, more particularly, to a method for estimating and thereafter adjusting contrast of a digital image.

Appendix

A typical digital imaging system captures an image and, after various automatic or interactive manipulations, displays the image on a monitor or prints a hardcopy of the image both for permitting human visual viewing. The perceived contrast of the printed or displayed image, as a pleasing reproduction of the original scene contrast, depends mainly on two factors: the scene contrast and the system reproduction contrast.

The scene contrast is affected by the scene illumination (the lighting contrast), the distribution of object reflectance factors (the object contrast), and the distribution of object color (the color contrast).

The system reproduction contrast is determined by the various steps along the imaging chain, such as sensor calibration, tone scale mapping, and output device characteristics. In order to achieve automatic contrast adjustment in digital photofinishing, one needs a method for estimating the scene contrast, and based on this estimate, a method for adjusting the reproduced image contrast or system contrast for optimal visual reproduction.

US-A-4,731,671 teaches a method for contrast adjustment in digital image processing. This method creates a plurality of predetermined Laplacian response intervals and then computes the Laplacian for each pixel in an input image. The code value for each pixel in the image is then placed in its corresponding Laplacian response interval for accumulating several code value histograms, one for each Laplacian response interval. It then selects the histogram whose shape is closest to a normal distribution The standard deviation of the selected histogram is taken as an estimate of the scene contrast. The standard deviation is used as an estimate of the scene contrast because of the correlation between the standard deviation and the scene contrast. That is, a high standard deviation corresponds to a high scene contrast and a low standard deviation corresponds to a low scene contrast.

The estimated contrast is then compared with a distribution of scene contrasts (that is, standard deviations from a plurality of scenes) pre-computed from a plurality of random sample of images. If the estimated contrast is higher than the population average, then the image is considered to have a higher than normal contrast and the system reproduction contrast is then adjusted lower so that the printed image will have an image contrast that is closer to the average contrast. If the estimated image contrast is lower than the average, then the system reproduction contrast is raised accordingly.

There are also methods for contrast adjustment that are based on histogram transformation. This class of methods does not explicitly estimate the scene contrast in the input image. The method simply assumes that an image histogram has to be in a certain shape and proceeds to transform it into the desired shape. The best known example is the histogram equalization method.

Still further, a theoretical construct is described by W.A. Richards, "Lightness Scale From Image Intensity Distribution," Applied Optics, 21, 14, pp. 2569-2582, 1982. The idea is that a "randomly" sampled log-exposure histogram of an image should have a shape similar to a normal distribution.

Although the presently known and utilized methods for adjusting tone scale in a digital image are satisfactory, they are not without drawbacks. The method of US-A-4,654,722 seems to perform rather well for most images; however, there are a few situations when there are shortcomings. First, there are two parameters that have to be predetermined: the lower threshold and the width of the Laplacian interval. The patent does not provide an automatic method for setting these two parameters. As a consequence, for some images, the lower threshold is not high enough to exclude noise and textures, therefore, causing the standard deviation of the selected histogram to be unduly biased by large uniform areas (when the noise is higher than the lower threshold) or by busy texture areas (such as grass or trees). Secondly, the selected histogram often exhibits bimodality for overcast scenes with sky in them. Although the scene contrast is low, the standard deviation of the selected histogram is large because of the bimodality caused by the dark grass pixels and the bright sky pixels.

In addition, the above-described histogram transformation method does not have a sound theoretical foundation, and frequently, it produces unacceptable tone reproduction for consumer images. Still further, in the method of W. A. Richards, a drawback arises in defining what constitutes a truly random sampling. Furthermore, another drawback of the histogram modification approach in the prior art is that the resulting tone transformation curve often has too high or too low local contrast in some portions of the curve. Therefore, the processed image does not look pleasing.

Consequently, a need exists for an improved method for estimating the scene contrast and adjusting the reproduc-

tion contrast based on this estimated scene contrast.

From the above discussion, there are three shortcomings in the prior art for image contrast estimation and adjustment: (1) large uniform areas can produce a bias on the contrast estimate, calculated over the entire image; (2) busy texture areas can produce a bias on the contrast estimate, calculated only on the edge pixels; and (3) contrast adjustment needs to be performed under constrained limits. In the present invention, a method of detecting the uniform areas and the busy texture areas in the input image is developed to solve the first two shortcomings and a method of constructing the final reproduction tone transformation curve within pre-specified contrast limits is used for adjusting the image contrast. A further advantage of this invention is that instead of transforming every image histogram into a fixed-shaped target histogram, such as a normal distribution, the present invention generates an image-dependent target histogram for each image. This greatly reduces the algorithm's sensitivity to sampling error.

In this invention, the method for estimating the scene contrast from a digital image is to sample only those high contrast pixels in the image and calculate the standard deviation of their log-exposure distribution. The computed standard deviation is taken as a measure of the scene contrast, which is used to compare with similarly calculated statistics from a large population of consumer images. If an input scene contrast is lower than the mean contrast of the precompiled population scene contrasts, then the contrast of the input image will be adjusted higher. If the input scene contrast is higher than the population mean contrast, then the input image contrast is slightly reduced.

In the present invention, the contrast adjustment is accomplished through a process commonly known as histogram modification. The original histogram is mapped (and thus modified) through a tone transformation curve into the target histogram. The present invention includes (1) a target histogram whose shape is derived from and therefore dependent on the original histogram computed from the input image, (2) a tone transformation curve that is constrained to have local slopes between the pre-imposed lower and upper bounds, and (3) a tone transformation curve that includes a fixed point in mid-tone range that maps a certain input code to the output code of the same value. In the preferred embodiment of the invention, this fixed point corresponds to the median of the input histogram and that of the output target histogram. Alternatively, this fixed point can be chosen by the users to maintain the desired white balance of the image.

Fig. 1 is a diagram illustrating a typical system for implementing the present invention;
Fig. 2 illustrates a flowchart of a software program of the present invention;
Fig. 3 illustrates in detail a portion of the software program and, more specifically, the procedure for estimating the scene contrast;
Fig. 4 illustrates in detail a portion of the software program and, more specifically, the procedure for determining thresholds in pixel sampling;
Fig. 5 illustrates an example of a typical Laplacian histogram, hist(L), of an image;
Fig. 6 illustrates the locations of the lower and upper thresholds as determined from the Laplacian histogram, hist(L), of Fig. 5;
Fig. 7 illustrates in detail a portion of the software program and, more specifically, sampling pixels from the original image;
Fig. 8 in detail a portion of the software program and, more specifically, the procedure for constructing the contrast-adjusted tone scale curve for the luminance signal;
Fig. 9 illustrates the function that is used to determine the standard deviation of the target histogram from the standard deviation of the input histogram; and
Fig. 10 illustrates the construction of the tone transformation curve y(x) from the input histogram K(x) and the target histogram Q(y).

In the below-described preferred embodiment, the computer readable storage medium may comprise, for example; magnetic storage media such as a magnetic disc (such as a floppy disc) or magnetic tape; optical storage media such as an optical disc, optical tape, or machine readable bar code; solid state electronic storage devices such as read only memory (ROM), or random access memory (RAM); or any other physical device or medium employed to store a computer program.

Referring to Fig. 1, there is illustrated a typical general purpose computer system 5 for implementing the software program of the present invention. The software program may be installed for execution on the general purpose computer system 5 via any suitable computer readable storage medium 10, as is well known in the art.

A captured image 20 is input into the general purpose computer 5 via any well known means, such as a scanner 30, where it is digitized and preferably converted into a red, green and blue (RGB) digital image format. The digitized image is transmitted to a central processing unit 40 where it is processed by the software program of the present invention. The digitized image is also preferably calibrated in log-exposure metric by the central processing unit 40; such calibration into log-exposure metric is well known in the art. The digitized image may also be displayed on a monitor 50 during or after processing.

Referring to Fig. 2, there is illustrated an overview of the software program of the present invention for estimating the input scene contrast and adjusting the output image contrast automatically. The software program is initiated S10, and the digital image is first processed with any well known color and density balance algorithm S20 to correct for color cast due to illumination and for exposure variation due to errors in the camera exposure control. The function of the color and density balance algorithm is for adjusting the red, green, and blue records of the input image so that some chosen digital code values are printed (or displayed) to a medium gray in the output media. For example, the digital code values (R,G,B) = (2000, 2000, 2000) are printed on the output photographic paper at status A densities (0.8, 0.8, 0.8).

After the image has been color and density balanced, any well known color transform is applied S30 to the balanced image so that the RGB signal is transformed into any suitable color space that has one (luminance) component representing the lightness variation and two (chrominance) components representing the chromatic variation in the image. The following color space is preferred:

$$\text{Luminance: } Cl=(R+G+B)/\sqrt{3}$$

$$\text{Chrominance: } Cs=(R-B)/\sqrt{2}$$

$$\text{Chrominance: } Ct=(2G-R-B)/\sqrt{6}$$

where R,G,B are the red, green, and blue log-exposures, although any other perception based space may also be used, such as CIELAB. It is instructive to note that in the present invention the contrast of the luminance component is adjusted independently from that of the chrominance component.

Next, the scene contrast is estimated S40 from the luminance component of the input image, as will be described in detail below in reference to Fig. 3. A tone scale curve is then constructed S50 from the estimated scene contrast, and the luminance image is then mapped S60 through the tone scale curve for adjusting the tone scale, as will also be described in detail below. The program may then be exited S70, and the digital image printed after the luminance image has been properly combined with the chrominance images by the following inverse transform:

$$R=Cl/\sqrt{3}+Cs/\sqrt{2}-Ct/\sqrt{6}$$

$$G=Cl/\sqrt{3}+2Ct/\sqrt{6}$$

$$B=Cl/\sqrt{3}-Cs/\sqrt{2}-Ct/\sqrt{6}$$

. Alternatively, the results of the program may be further processed by other well known image processing programs for further processing the image.

Referring to Fig. 3, there is illustrated the step of estimating the scene contrast S40. First, two thresholds (the lower and the upper thresholds) are determined S80 for a Laplacian response, as will be illustrated in detail in Fig. 4. Then, the two thresholds are used in deciding which pixels should be sampled from the digital image for further analysis in eventually determining the tone scale curve. A histogram K(x) derived from the sampled pixels is created S100 and its standard deviation (k) is calculated S110, where x denotes the input code value. The output k is used as an estimate of the scene contrast. Output k is used as an estimate of the standard deviation because of the previously stated correlation between the standard deviation and the scene contrast.

Referring to Fig. 4, there are illustrated the details of the step for determining the thresholds S80. The well known Laplacian operator is applied S120 to the input luminance image and the Laplacian response, L, for each pixel is calculated S130 to form the Laplacian histogram, hist(L). A typical Laplacian kernel is illustrated in block S120, although any Laplacian kernel may be used. (see *The Image Processing Handbook*, Second Edition, pages 225-232, for further discussion)

The Laplacian histogram, hist(L), can be well approximated by a mixture model:

$$h(L)=\frac{a}{\sqrt{2\pi}\sigma}e^{-\frac{L^2}{2\sigma^2}}+\frac{1-a}{2\lambda}e^{-\frac{|L|}{\lambda}},$$

where $0 \le a \le 1$, σ is the standard deviation of the Gaussian component, and λ is the decay constant of the double-exponential component. Since the first term represents pixels from uniform and texture areas, the thresholds should be selected at substantially $\pm 2\sigma$ to $\pm 3\sigma$. It is possible to determine the parameter σ from a least square fit of the model, but

a more efficient approximation is to take the second derivative of hist(L) S140, and find the local maxima along the negative and the positive axis, as described in detail below.

Mathematically, we want to locate zeroes in the third derivative:

$$\frac{d^3h}{dL^3} = \frac{a}{\sqrt{2\pi\sigma^5}} L\left(3 - \frac{L^2}{\sigma^2}\right) e^{-\frac{L^2}{2\sigma^2}} - \frac{1-a}{2\lambda^4} e^{-\frac{|L|}{\lambda}} = 0$$

For consumer images, σ is much smaller than λ, and the zeroes are approximately located at $d_1 \approx -\sqrt{3}\sigma$ and $d_2 \approx +\sqrt{3}\sigma$. Therefore, in the preferred embodiment of the invention, the thresholds, t_1 and t_2, are selected S145 at $t_1=f(d_1)=1.5d_1$ and $t_2=f(d_2)=1.5d_2$, that is at approximately $\pm 2.6\sigma$, which will exclude 99% of all the pixels from the uniform and texture areas and will exclude 99% of the pixels containing noise.

Fig. 5 illustrates an example of the Laplacian histogram of a consumer image. The circles are the data points, the thick solid curve is the mixture model, and the two thin, dashed curves are the two components of the mixture model. Fig. 6 depicts where the lower and the upper thresholds, t_1 and t_2, are located by the method of maximal second derivative.

After the two thresholds are determined, Fig. 7 illustrates how the pixels are sampled and the histogram K(x) is computed S100. For each pixel with a code value (x) in the input image S150, its Laplacian value is computed S160. If its Laplacian value is greater than the upper threshold or less than the lower threshold S170, then the pixel is retrieved or sampled and the histogram K(x) is incremented S180 by 1 at code value x. This sampling and incrementing is repeated for the entire image S190. After the calculation of the histogram K(x) is completed, the standard deviation, k, of K(x) is then computed as an estimate of the scene contrast.

Referring to Fig. 8, once the scene contrast estimate, k, is computed, the tone scale is computed S50. First, a target histogram, Q(y), is computed and thereafter the desired tone transformation curve y(x) using a finite-difference stepwise construction procedure is constructed, where y denotes the output code value.

The target histogram, Q(y), is computed by convolving the sample histogram K(y) with a Gaussian distribution G(y), and then by scaling the resulting function with a scaling factor, s. The standard deviation g of G(y) determines the shape of the target function. If g >> k, then the target function Q(y) is essentially a Gaussian distribution. In this case, the tone transformation curve, y(x), will map K(x) to a Gaussian distribution. If g << k, then the target function Q(y) will have essentially the same shape as K(x). In this case, the tone transformation curve, y(x), is essentially a straight line. Its slope determines the global contrast adjustment. Therefore, the single parameter, g, allows the user to control the desired manipulation of the characteristics of the tone transformation curve. In the general case, g is adjusted according to the shape parameters of the histogram K(x) by a function p(k) S180. It is instructive to note that other factors may be used as arguments of p(k) in the determination of g. For example, if the skewness of K(x) is large, the value of g can be made small, so that the target histogram, Q(y), also has a large skewness. In the simplest implementation, g is made to be a function, p(k), of the scene contrast estimate, k, alone. Experimental results show that a simple function, g=p(k)=k is sufficient to produce very pleasing images. Once the standard deviation g is determined, the function G can be constructed S190. This choice of g=k produces a target function Q(y) that is halfway between K(x) and a Gaussian distribution as a result of the convolution of K(y) and G(y) S200.

The variance of the convolution product, F(y), of K(y) and G(y) is the sum of the variances of K(y) and G(y), that is, $k^2 + g^2$. Consequently, F(y) has a wider spread than the input sampled histogram K(x). If the goal is only to change the shape, but not the spread, of the histogram, then F(y) has to be scaled down back to the same variance as that of K(x). However, the goal of digital printing is to adjust the image contrast to produce a pleasing reproduction of the input image. This is achieved by adjusting the standard deviation, c, of the target histogram. For this purpose, a scaling factor S210,

$$s = \sqrt{\frac{c * c}{k * k + g * g}},$$

is used to control the final spread of the target function, Q(y). It is used to scale the convolution product, F(y), to get Q(y):

$$Q(y) = \frac{1}{s} F\left(\frac{y}{s}\right).$$

Referring to Fig. 9, the standard deviation, c, of the target histogram Q(y), is determined from a function u of the standard deviation of the input histogram K(x). If $c = k$, then the target function Q(y) will have the same standard deviation as K(x), resulting in no adjustment of global contrast. If c is chosen to be greater than k, then the global contrast will be increased. Therefore, if we make c a function of k, that is, c=u(k) S205, then we can adaptively adjust the global contrast of the input image. For example, if k is smaller than the population mean contrast, k_0, then we should make c greater than k and thereby increase the image contrast. In the preferred embodiment of this invention, the function c=u(k) is implemented as a one-dimensional lookup table with the following functional form: $c = k_0(1-\exp[-k/k_1]+\exp[-k_0/k_1])$, when $k < k_0$, and $c=k_0+b^*(k-k_0)$, when $k \geq k_0$, where k_0, k_1 and b are constants determined from population statistics.

Referring to Fig. 8 and 9, they illustrate the construction of the tone transformation curve y(x) from the input histogram K(x) and the target histogram Q(y). The procedure to construct the curve y(x) that maps one histogram, K(x), to another histogram, Q(y), is defined by the following equation:

$$\int_{-\infty}^{y} Q(\xi)d\xi = \int_{-\infty}^{x} K(\eta)d\eta.$$

Taking derivative on both sides, we have: $Q(y)dy = K(x)dx$, which can be solved by a finite-difference method with a boundary condition. Since the color and density balanced input image has a defined balanced point, say x_0, which is usually the point that will be mapped to the medium gray at the output, we can set the boundary condition of y(x) so that $y_0=y(x_0)=x_0$. This boundary condition requires that the target distribution Q(y) be translated along the y axis so that

$$\int_{-\infty}^{y=x_0} Q(\xi)d\xi = \int_{-\infty}^{x_0} K(\eta)d\eta.$$

Therefore, the density balanced point is not changed by the tone transformation curve. However, if the output image is to be re-balanced again, then the particular selection of the boundary condition is not very important. Defining the slope function of y(x), $S(x,y) = K(x)/Q(y)$, the finite-difference solution of y as a function of x is given S220 by:

$$y(x_{1+i})=y(x_i)+S(x_i,y(x_i))(x_{1+i}-x_i),$$

starting with $y(x_1)=y(x_0)+S(x_0,y(x_0))(x_1-x_0)$ where $y(x_0)=x_0$.

The tone transformation curve so constructed may not be very smooth because of the discrete implementation in digital computers. In order to get a smooth curve, it may be desirable to smooth the curve by a low-pass filter S230, such as a Gaussian filter. The finite-difference method of constructing the tone transformation curve provides an easy way to impose the limits, MAXIMUM SLOPE and MINIMUM SLOPE, on the instantaneous slope $S(x_i,y(x_i))$. If at any step, the slope $S(x_i,y(x_i))$ exceeds the limits, then its value is clipped at the limits. Therefore, the resulting tone transformation curve is guaranteed to have its slope vary within the specified limits. This is a very important property of a good, practical tone transformation.

Other features of the invention are included below.

The method further comprising the step of computing a tone scale transformation curve from the estimate contrast of the digital image.

The method further comprising the step of constructing the tone transformation curve by modifying the histogram formed from the sampled pixels so that a target histogram is generated which the target histogram is dependent on the digital image.

The method further comprising the step of constructing a tone transformation curve by convolving the sampled pixel histogram with another pre-defined function with image-dependent parameters.

The method wherein the pre-defined function is a Gaussian distribution.

The method further comprising forming a target histogram from the sampled histogram by convolving the sampled histogram with a pre-defined function, and constructing a tone scale curve by transforming the sampled histogram to the target histogram.

The method wherein the step of determining the first and second thresholds includes finding a local maxima of a second derivative of the Laplacian histogram.

The method wherein the step of selecting the pixels includes selecting the pixels that exceed substantially the first threshold.

The method wherein the step of selecting the pixels includes selecting the pixels that are less than substantially the

second threshold.

The method further comprising the step of computing a standard deviation of the sampled histogram and estimating contrast of the digital image by comparing the computed standard deviation with a predetermined contrast for determining contrast of the input image in relationship with the predetermined contrast.

The method wherein the step of determining the first and second thresholds includes finding a local maxima of a second derivative of the Laplacian histogram.

The method wherein the step of selecting the pixels includes selecting the pixels that exceed substantially the first threshold.

The method wherein the step of selecting the pixels includes selecting the pixels that are less than substantially the second threshold.

The method further comprising the step of constructing the tone transformation curve by modifying the histogram formed from the sampled pixels so that an image-dependent target histogram is generated.

APPENDIX


```
/* contrast_tone.c -- use the Lapacian histogram to select edge pixels
              and build a tone scale curve with histogram modification */
/* output a lookup table */
/* Hsien-Che Lee, Nov. 18, 1996 */
/* cc -o contrast_tone contrast_tone.c -lm */

#include <math.h>
#include <string.h>
#include <stdio.h>

#define RECLNG  80
#define SLOPE_MIN 0.9
#define SLOPE_MAX 1.6
#define BLEND 1.0
#define MAXCODE 4095

short   *red, *green, *blue;

main(argc,argv)
int     argc;
char    *argv[];
{
char    *imginname, histname[80];
unsigned int    picsize, hist_size;
short   *rr, *gg, *bb, *lum, *lu, *lap, *la;
short   *nn, *ee, *ss, *ww, *cc, code;
int     pixels, lines, pixel, line, inc, *hist, id;
int     i, k, j, offset, r, l, m, n, *lut, *nhist, pivot;
int     pthreshold, nthreshold, new_size, block, xpivot, ypivot;
int     raim, gaim, baim, rshift, gshift, bshift, aim, naim;
float   a, b, count, p, ptile, s, sigma, yy, xsum, y, c, c1, c2;
float   amax, threshold, *fhist, *inhist, *outhist, laplacian;
void    read_img(), smooth(), compute_target_histogram();
FILE    *fd, *fout, *fo;

if(argc != 5)   {
    printf("Usage: contrast_tone imagein rshift gshift bshift\n");
    exit(1);
    }
else
    {
    argv++;
    imginname = *argv++;
    rshift = atoi(*argv++);
    gshift = atoi(*argv++);
    bshift = atoi(*argv++);
    }

/* default aims */
raim = 1550; gaim = 1550; baim = 1550;
aim = (raim + gaim + baim) / 3;

read_img(imginname, &pixels, &lines);

/* form the luminance channel */
picsize = pixels * lines;
lum = (short *) calloc(picsize, sizeof(short));
rr = red; gg = green; bb = blue; lu = lum;
offset = (rshift + gshift + bshift) / 3;
for(i = 0; i < picsize; i++) {
        *lu++ = ((*rr++) + (*gg++) + (*bb++)) / 3 + offset;
        }
free(red); free(green); free(blue);

hist_size = 5000;
if(hist_size <= MAXCODE) hist_size = MAXCODE + 1;
```

```
offset = hist_size / 2;
block = 10;
new_size = hist_size / block;
hist = (int *) calloc(hist_size, sizeof(int));
lut = (int *) calloc(hist_size, sizeof(int));
nhist = (int *) calloc(new_size, sizeof(int));
fhist = (float *) calloc(hist_size, sizeof(float));
inhist = (float *) calloc(hist_size, sizeof(float));
outhist = (float *) calloc(hist_size, sizeof(float));

/* calculate the Laplacian histogram */
lap = (short *) calloc(picsize, sizeof(short));
pixel = pixels - 1; line = lines - 1;
inc = 4; c = 0.0; la = lap + pixels + 1;
nn = lum + 1;
ww = lum + pixels; cc = ww + 1; ee = cc + 1;
ss = cc + pixels;
for(j = 1; j < line; j++) {
   for(i = 1; i < pixel; i++) {
        laplacian = (*ee++)+(*ww++)+(*nn++)+(*ss++)-4*(*cc++);
        *la++ = laplacian;
        k = (int) (laplacian + offset + 0.5);
        if(k < 0) continue;
        if(k >= hist_size) continue;
        hist[k] += 1; c+= 1.0;
        }
   la += 2; nn += 2; ww += 2; cc += 2; ee += 2; ss += 2;
   }

/* find the maximal 2nd derivative and use it to determine
   the threshold */
for(i = 0; i < hist_size; i++) fhist[i] = (float) hist[i] ;
smooth(fhist, hist_size, 10.0);
amax = -1000.0;
for(i = offset + 1; i < hist_size - 2; i++) {
        c = fhist[i-2] + fhist[i+2] - fhist[i] - fhist[i];
        if(c > amax) {amax = c; threshold = i - offset;}
        }
printf("threshold = %.1f\n", threshold);

pthreshold = threshold * 1.5;
nthreshold = -threshold * 1.5;

compute_hist:

/* calculate the input histogram from sampled pixels */
for(i = 0; i < hist_size; i++) hist[i] = 0;
pixel = pixels - 1; line = lines - 1;
c = 0.0;
la = lap + pixels + 1;
cc = lum + pixels + 1;
for(j = 1; j < line; j++) {
        for(i = 1; i < pixel; i++) {
                k = *la++;
                if(k > pthreshold || k < nthreshold) {
                        code = *cc;
                        hist[code]++;
                        c += 1.0;
                        }
                cc++;
                }
        la += 2; cc += 2;;
        }
free(lap);

/* if there are too few pixels, lower the thresholds
```

```
   and try again */
if(c < 1000.0) {
        pthreshold *= 0.9;
        nthreshold *= 0.9;
        goto compute_hist;
        }

/* find the 50 percentile point */
ptile = 0.5;
c2 = c * ptile; i = 0; xpivot = -10.0;
a = 0;
while(xpivot < 0) {
        a += hist[i];
        if(a >= c2) xpivot = i;
        i++;
        }

/* form a coarse histogram */
k = 0;
for(i = 0; i < new_size; i++) {
        xsum = 0.0;
        for(j = 0; j < block; j++){
                xsum += hist[k++];
                }
        nhist[i] = xsum;
        }

xpivot /= block;
c = 0.0;
for(i = 0; i < new_size; i++) {
        fhist[i] = (float) nhist[i];
        c += nhist[i];
        }
smooth(fhist, new_size, 3.0);
c = 0.0;
for(i = 0; i < new_size; i++) {
        nhist[i] = (int) fhist[i];
        c += nhist[i];
        }

/* compute the target histogram, hist, by convolving the
   input histogram, nhist, with a Gaussian function */

compute_target_histogram(nhist, hist, new_size, xpivot);

/* find ypivot */
c = 0.0;
for(i = 0; i < new_size; i++) c += hist[i];
c2 = c * ptile; i = 0; ypivot = -10.0;
a = 0;
while(ypivot < 0) {
        a += hist[i];
        if(a >= c2) ypivot = i;
        i++;
        }

/* normalize the input histogram and the target histogram */
c1 = 0.0; c2 = 0.0;
for(i = 0; i < new_size; i++) {c1 += nhist[i]; c2 += hist[i];}
for(i = 0; i < new_size; i++) {
        inhist[i] = nhist[i] / c1;
        outhist[i] = hist[i] / c2;
        }

/* construct the tone scale curve that maps the input histogram
   to the output target histogram, outhist, under the constraint
   that the local slope of the tone scale curve should not exceed
   SLOPE_MIN and SLOPE_MAX. Also the tone scale curve will map
   the aim point (1550) to the same code value in the output. */
/* The tone scale curve is first constructed on the coarser
   scale fhist[0-255], and then interpolated to the fine scale
   lut[0-4095] */

id = ypivot - xpivot;
fhist[xpivot] = xpivot;

/* construct the upper part of the tone scale curve */
c1 = inhist[xpivot];
c2 = outhist[ypivot];
c = c1 / c2;
yy = ypivot;
for(i = xpivot + 1; i < new_size; i++) {
        if(c < SLOPE_MIN) c = SLOPE_MIN;
        if(c > SLOPE_MAX) c = SLOPE_MAX;
        y = yy + c;
        if(y < 0) y = 0.0;
        if(y > new_size) y = new_size;
        j = (int) (y + 0.5);
        n = j - id;
        if(n < 0) n = 0;
        if(n >= new_size) n = new_size - 1;
        fhist[i] = n;
        yy = y;
        c1 = inhist[i];
        c2 = outhist[j];
        if(c2 == 0) {c = 1.0;} else {c = c1 / c2;}
        }

/* construct the lower part of the tone scale curve */
c1 = inhist[xpivot];
c2 = outhist[ypivot];
c = c1 / c2;
yy = ypivot;
for(i = xpivot - 1; i >= 0; i--) {
        if(c < SLOPE_MIN) c = SLOPE_MIN;
        if(c > SLOPE_MAX) c = SLOPE_MAX;
        y = yy - c;
        if(y < 0) y = 0.0;
        if(y > new_size) y = new_size;
        j = (int) (y + 0.5);
        n = j - id;
        if(n < 0) n = 0;
        if(n >= new_size) n = new_size - 1;
        fhist[i] = n;
        yy = y;
        c1 = inhist[i];
        c2 = outhist[j];
        if(c2 == 0) {c = 1.0;} else {c = c1 / c2;}
        }

/* smooth out the tone scale curve */
smooth(fhist, new_size, 4.0);

/* interpolate the tone scale to finer resolution */
for(i = 0; i < hist_size; i++) {
        j = i / block;
        k = j+1; if(k >= new_size) k = new_size - 1;
        a = fhist[j];
        b = fhist[k];
        c = (float) (i-j*block)/(float) block;
        p = a * (1.0 - c) + b * c;
        lut[i] = (int) (p*block+0.5);
```

```c
        }

/* shift lut so that aim is mapped to aim */
id = aim - lut[aim];
for(i = 0; i < hist_size; i++) {
        k = lut[i] + id;
        if(k > MAXCODE) k = MAXCODE;
        if(k < 0) k = 0;
        lut[i] = k;
        }

/* write out the tone scale lookup table */
sprintf(histname, "bqd_lut.dat", imginname);
if((foute fopen(histname,"w")) == NULL) {
   printf("Cannot open %s\n", histname);
   exit(1);
   }
for(i = 0; i <= MAXCODE; i++) {
   fprintf(fout, "%d %d %d %d\n", i, lut[i], i-240, i-240);
   }
fclose(fout);

}

/* read in the input image */
void
read_img(imginname, pix, lin)
char    *imginname;
int     *pix, *lin;
{
char    doc[128], rimg[128], gimg[128], bimg[128];
int     bytes, pixels, lines, picsize;
FILE    *fd, *fdr, *fdg, *fdb;

/* assuming two bytes per pixel */
bytes = 2;

sprintf(doc, "%s.doc", imginname);
if((fd= fopen(doc,"r")) == NULL) {
        printf("Cannot open %s\n", doc);
        exit(1);
        }
fscanf(fd, "%d %*s", &lines);
fscanf(fd, "%d %*s", &pixels);
fclose(fd);
printf("pixels= %d lines= %d\n", pixels, lines);

picsize = pixels * lines;

red = (short *) calloc(picsize, sizeof(short));
green = (short *) calloc(picsize, sizeof(short));
blue = (short *) calloc(picsize, sizeof(short));

sprintf(rimg, "%s.red", imginname);
sprintf(gimg, "%s.grn", imginname);
sprintf(bimg, "%s.blu", imginname);

if((fdr= fopen(rimg,"r")) == NULL) {
        printf("Cannot open %s\n", rimg);
        exit(1);
        }
fread((char *) red, bytes, picsize, fdr);
fclose(fdr);

if((fdg= fopen(gimg,"r")) == NULL) {
        printf("Cannot open %s\n", gimg);
        exit(1);
        }
fread((char *) green, bytes, picsize, fdg);
fclose(fdg);

if((fdb= fopen(bimg,"r")) == NULL) {
        printf("Cannot open %s\n", bimg);
        exit(1);
        }
fread((char *) blue, bytes, picsize, fdb);
fclose(fdb);

*pix = pixels; *lin = lines;
return;
}

void
smooth(hist, m, sigma)
/* this function applied a Gaussian smoothing filter, with a
   standard deviation equal to sigma, to the input histogram,
   hist, which is m element long */
/* the input histogram is replaced by its smoothed version */
float   *hist, sigma;
int     m;  /* m is the number of bins in the histogram hist */
{
int     i, j, n, nn, conv_size;
float   sq2pi, aa, s2, x, *wt, *kk, *h, sum;

sq2pi=2.506628;
/* mask is 4 sigma on each side */
conv_size = (int) (sigma * 4.0 + 0.5);
nn = 2 * conv_size + 1;
wt = (float *) calloc((unsigned) nn, sizeof(float));
n = nn - 1;
aa = (1.0 / sq2pi) / sigma;
s2 = -0.5 / sigma / sigma;
x = -n / 2.0 - 1.0;
/* calculate the Gaussian convolution kernel, wt */
/* coefficient aa is calculated so that the sum of
   the kernel is 1.0 */
for(i = 0; i < nn; i++) {
        x = x + 1.0;
        wt[i] = aa * exp((double) s2 * x * x);
        }

/* copy the input histogram to a suitably-sized array
   for convolution */
n = m + nn;
h = (float *) calloc((unsigned) n, sizeof(float));
for(i = 0; i < m; i++) {h[i+conv_size] = hist[i];}

/* extend two ends by the end values*/
for(i = 0; i < conv_size; i++) {
        h[i] = h[conv_size];
        h[m+conv_size+i] = h[m-1+conv_size];
        }

/* smooth the input histogram h and store it back in hist */
for(i = 0;  i < m; i++) {
        sum = 0.0;
        kk = &h[i];
        for(j = 0; j < nn; j++) sum += *kk++ * wt[j];
        hist[i] = sum;
        }
free(wt);
free(h);
```

```
}

/* compute the target histogram by convolving the
   input histogram with a Gaussian function */
void
compute_target_histogram(ihist, ohist, histsize, pivot)
int     *ihist;         /* input histogram */
int     *ohist;         /* output target histogram */
int     histsize;       /* dimension of fhist and hist */
int     pivot;          /* pivot point for contrast adjustment */
{
int     i, j, k, m, ypivot;
float   s, p, b, sigma_in, sigma_gauss, sigma_out, scale;
float   sum, ptile, c2, *fhist;
double  count, x1, x2, x, y, tt, m1, m2;
void    sigma_in_sigma_out();

/* calculate moments */
x1 = 0.0; x2 = 0.0; count = 0.0;
for(i = 0; i < histsize; i++) {
        x = (float) i;
        y = ihist[i];
        count += y;
        tt = x * y;
        x1 += tt;
        tt *= x;
        x2 += tt;
        }
m1 = x1 / count;
m2 = x2 / count - m1 * m1;
sigma_in = sqrt(m2);

sigma_gauss = BLEND * sigma_in;
if(sigma_gauss < 1.0) sigma_gauss = 1.0;

k = (int) (sigma_gauss * 4.0 + 0.5); /* mask is 4 sigma on each side */
m = histsize + 2 * k;
fhist = (float *) calloc(m, sizeof(float));
for(i = 0; i < k; i++) {fhist[i] = 0.0; fhist[m-i] = 0.0;}
for(i = 0; i < histsize; i++) {fhist[i+k] = (float) ihist[i];}

/* convolve the input histogram with a Gaussian histogram */
smooth(fhist, m, sigma_gauss);

/* fhist now contains the desired target histogram, except that
   it needs to be scaled back to get the right contrast */
/* the required ypivot point and scaling factor are calculated below */

/* find the percentile at the pivot point */
sum = 0.0;
for(i = 0; i <= pivot; i++) sum += ihist[i];
ptile = sum / count;

/* find ypivot */
count = 0.0;
for(i = 0; i < m; i++) count += fhist[i];
c2 = ptile * count; i = 0; ypivot = -10.0;
sum = 0;
while(ypivot < 0) {
        sum += fhist[i];
        if(sum >= c2) ypivot = i;
        i++;
        }

/* determine the required scaling factor */
b = sigma_in * sigma_in / (sigma_in * sigma_in
    + sigma_gauss * sigma_gauss);

/* factor needed to scale back to original sigma_in */
scale = sqrt(b);

/* contrast adjustment */
/* for sigma_in, determine sigma_out */
sigma_in_sigma_out(sigma_in, &sigma_out);

s = sigma_out / sigma_in;
printf("contrast adjustment factor: %f\n", s);
s *= scale; /* contrast adjustment times the scaling factor */

/* scale the target histogram to achieve the final contrast
   adjustment. */
for(i = 0; i < histsize; i++) {
        j = ypivot + (i - ypivot)/s + k;
        if(j < 0) j = 0;
        ohist[i] = (int) fhist[j];
        }
return;
}

void
sigma_in_sigma_out(sigma_in, sigma_out)
float   sigma_in, *sigma_out;
{
float   sigma, mean, upper, spread, a;

/* assume that sigma_in was computed from 100 * log E */
mean = 37.5; upper = 40.0; spread = 13.0;
/* if sigma_in was computed from 100 * printing density */
mean *= 0.65; upper *= 0.65; spread *= 0.65;

a = sigma_in; sigma = a;
if(a > upper) sigma = upper + 0.8 * (a-upper);
if(a < mean) {
   sigma = mean * (1.0 - exp(-a/spread) + exp(-mean/spread));
   }
*sigma_out = sigma;

return;
}
```

Claims

1. A method of estimating the scene contrast from a digital image, the method comprising the steps of:

   a) forming a Laplacian histogram distribution;
   b) determining, from the Laplacian histogram, first and second thresholds which eliminate substantially uniform areas or substantially textured portion of the digital image;
   c) selecting pixels which are based on the first and second thresholds from the digital image;
   d) forming a histogram from the sampled pixels;
   e) computing a standard deviation of the sampled histogram; and
   f) estimating contrast of the digital image by comparing the computed standard deviation with a predetermined contrast for determining contrast of the input image in relationship with the predetermined contrast.

2. The method of claim 1, wherein the step of determining the first and second thresholds includes finding a local maxima of a second derivative of the Laplacian histogram.

3. The method of claim 1, wherein the step of selecting the pixels includes selecting the pixels that exceed substantially the first threshold.

4. The method of claim 1, wherein the step of selecting the pixels includes selecting the pixels that are less than substantially the second threshold.

5. A method of estimating the scene contrast from a digital image, the method comprising the steps of:

   a) forming a Laplacian histogram distribution;
   b) determining, from the Laplacian histogram, first and second thresholds;
   c) selecting pixels which are based on the first and second thresholds from the digital image;
   d) forming a histogram from the sampled pixels;
   e) forming a target histogram from the sampled histogram by convolving the sampled histogram with a predefined function; and
   f) constructing a tone scale curve by transforming the sampled histogram to the target histogram.

6. The method as in claim 13, wherein the step of forming a target histogram includes convolving the sampled histogram with a Gaussian distribution.

7. The method as in claim 14, wherein the step of forming a target histogram includes convolving the sampled histogram with the Gaussian distribution having a standard deviation substantially the same as the sampled histogram.

8. A finite-difference method for constructing a transformation curve, y(x) that maps an input histogram K(x) to an output target histogram Q(y), so that the instantaneous slope of the curve is within specified limits, the method comprising the steps of:

   a) choosing a fixed point (x_0, y_0) where x_0 corresponds to a predetermined percentile of the input histogram K(x) and y_0 corresponds to the same predetermined percentile of the output target histogram Q(y);
   b) defining a slope function, S(x,y), as the ratio of the input histogram K(x), to that of the target histogram Q(y); and
   c) computing the transformation curve, step by step, for N points on the input histogram, x_i, i= 1 to N, using the equation: $y(x_{i+1}) = y(x_i) + S(x_i, y(x_i))\ (x_{i+1} - x_i)$, starting with $y(x_1) = y(x_0) + S(x_0, y(x_0))\ (x_1 - x_0)$ where $y(x_0)=x_0$; at each step, the value of the slope function, $S(x_i, y(x_i))$ is maintained between the specified limits.

9. A method for excluding pixels from a uniform portion or a texture portion of a digital image, comprising the steps of:

   a) forming a Laplacian histogram distribution;
   b) determining, from the Laplacian histogram, first and second thresholds which eliminate substantially the uniform portion or substantially the textured portion of the digital image; and
   c) selecting pixels which are based on the first and second thresholds from the digital image.

10. The method of claim 22, wherein the step of determining the first and second thresholds includes finding a local

maxima of a second derivative of the Laplacian histogram.

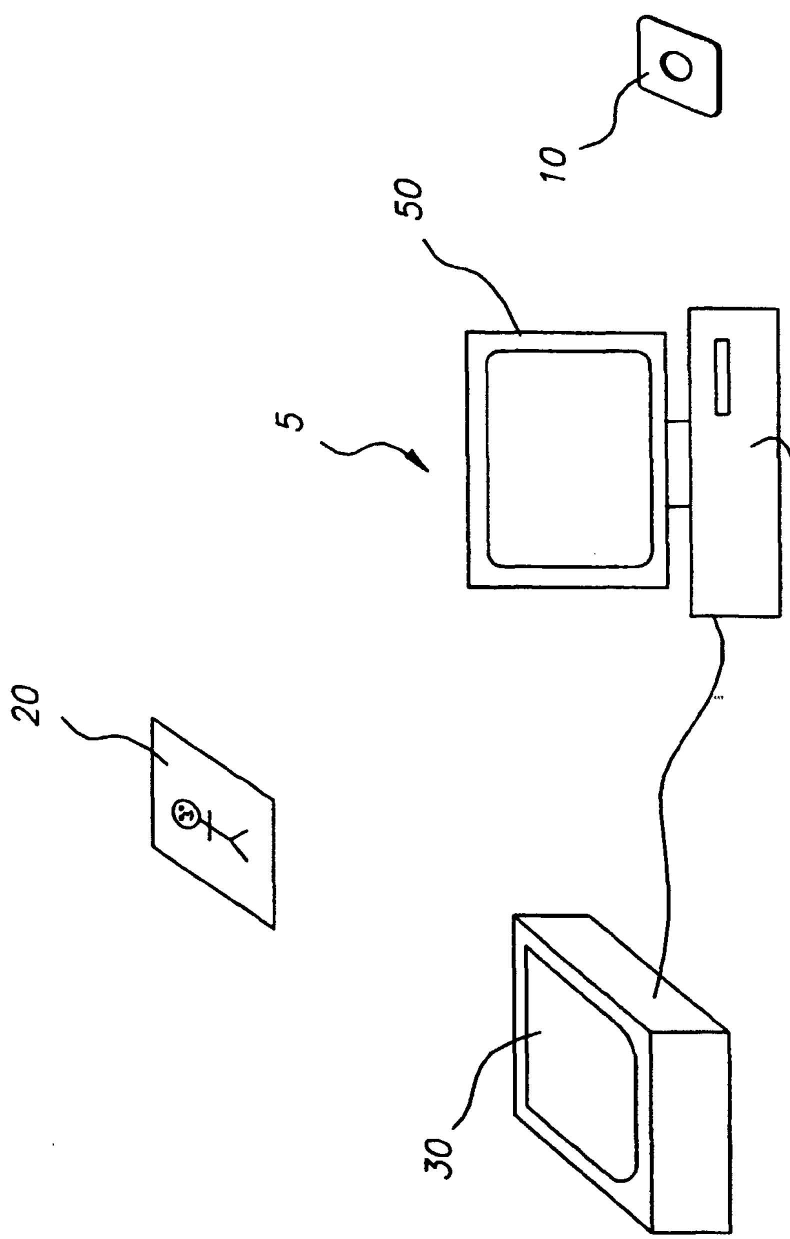

FIG. 1

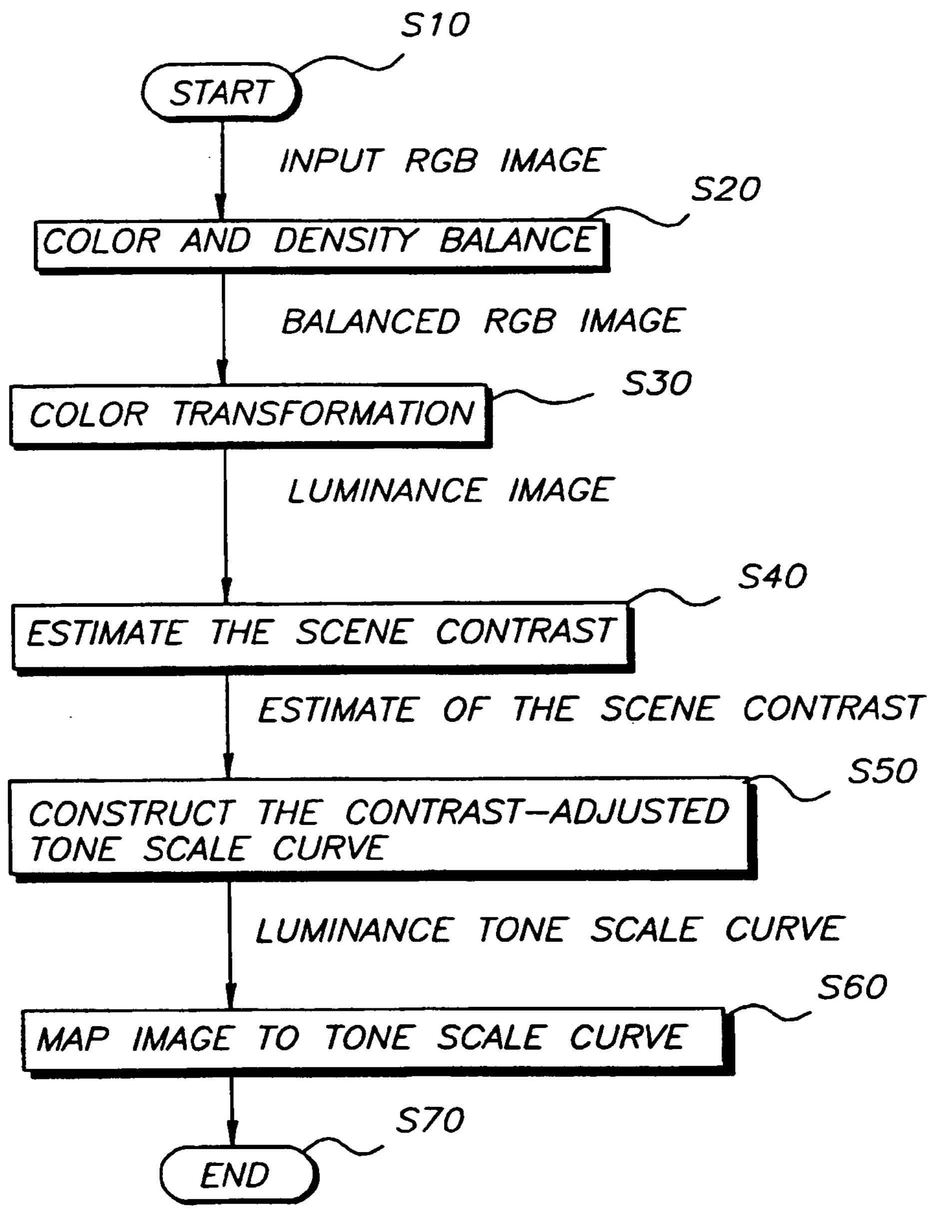
S10
START
INPUT RGB IMAGE
S20
COLOR AND DENSITY BALANCE
BALANCED RGB IMAGE
S30
COLOR TRANSFORMATION
LUMINANCE IMAGE
S40
ESTIMATE THE SCENE CONTRAST
ESTIMATE OF THE SCENE CONTRAST
S50
CONSTRUCT THE CONTRAST-ADJUSTED TONE SCALE CURVE
LUMINANCE TONE SCALE CURVE
S60
MAP IMAGE TO TONE SCALE CURVE
S70
END

FIG. 2

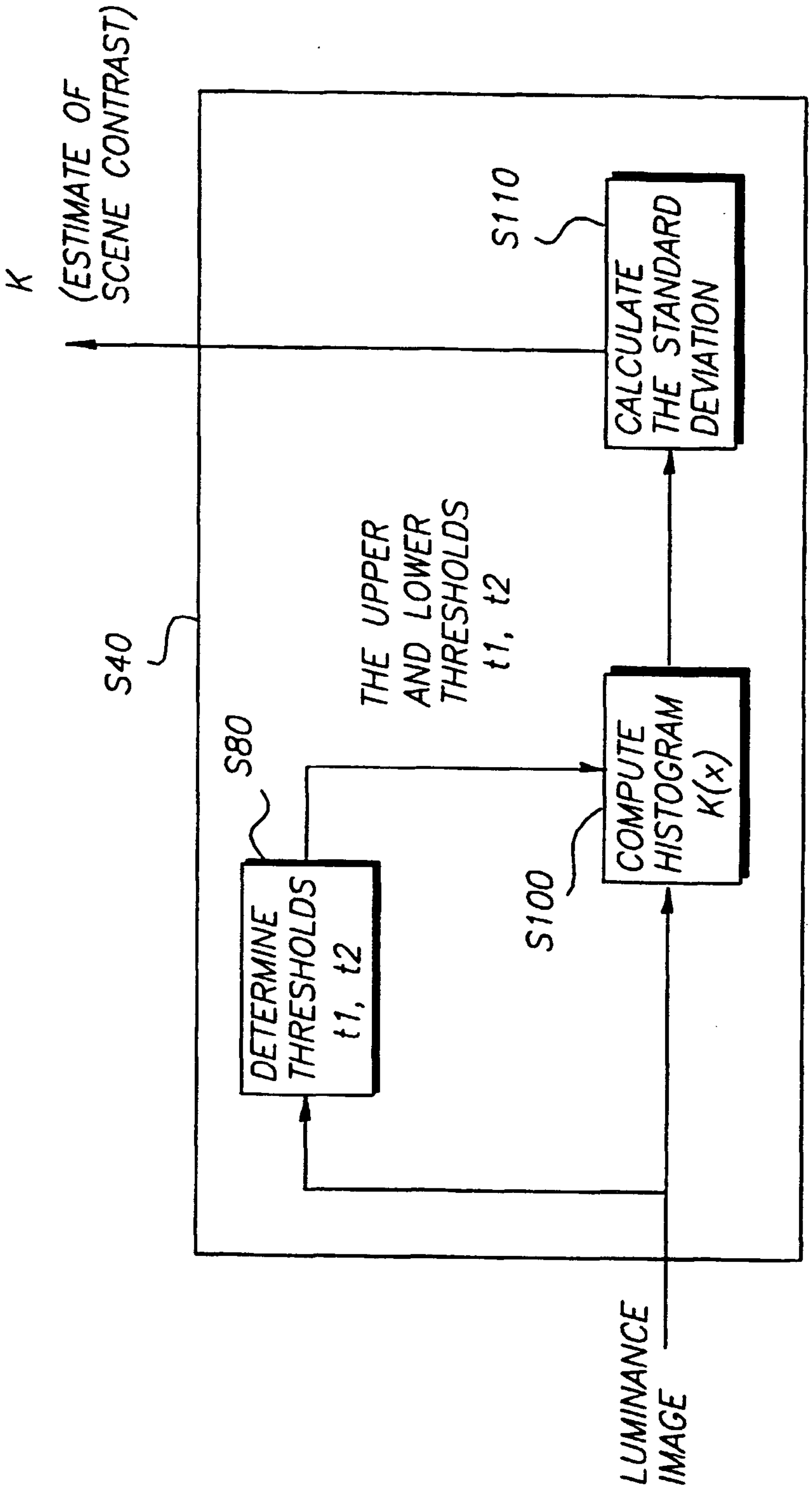
K
(ESTIMATE OF
SCENE CONTRAST)
S40
S80
DETERMINE
THRESHOLDS
t1, t2
THE UPPER
AND LOWER
THRESHOLDS
t1, t2
S100
COMPUTE
HISTOGRAM
K(x)
S110
CALCULATE
THE STANDARD
DEVIATION
LUMINANCE
IMAGE

FIG. 3

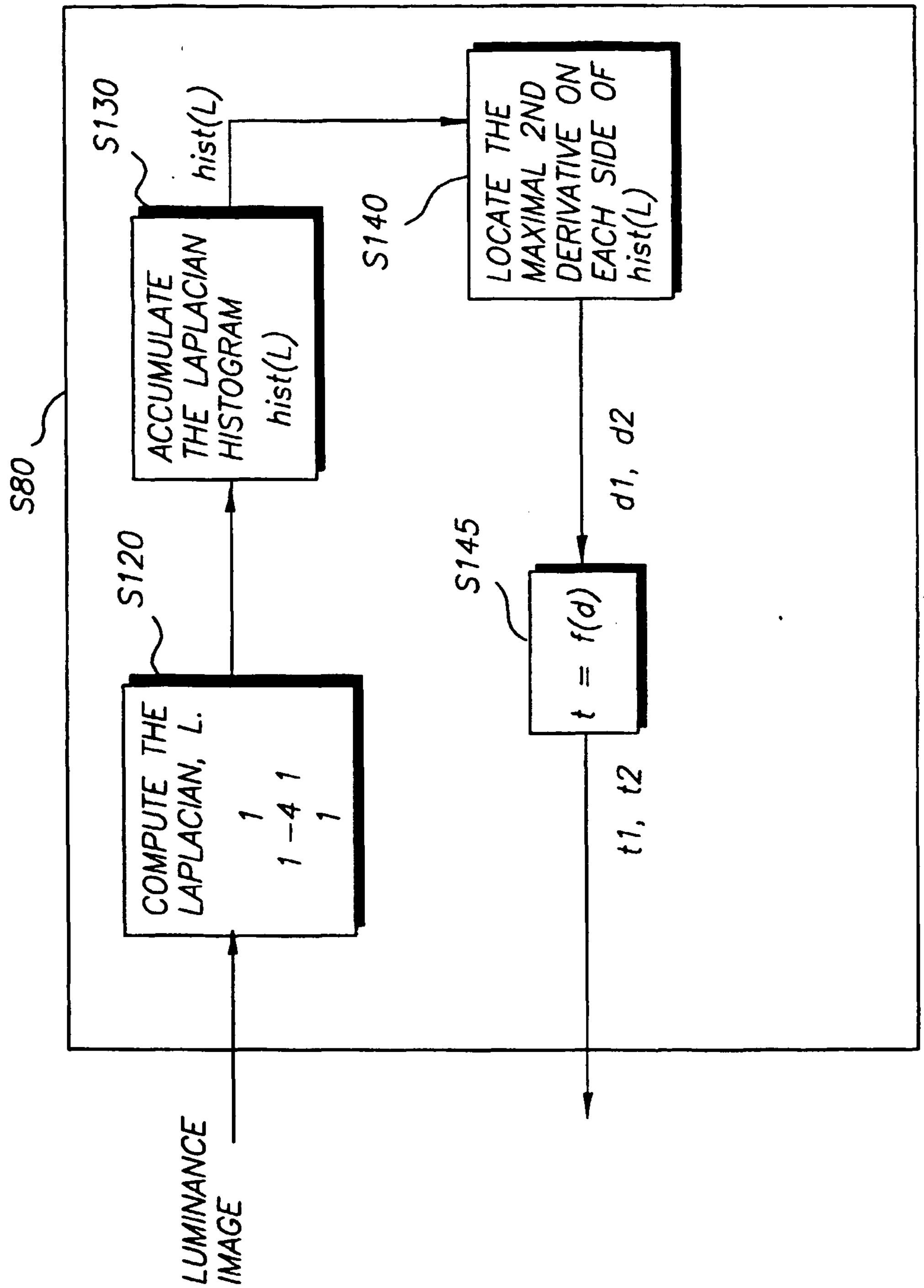
S80
S120
S130
S140
S145
LUMINANCE IMAGE
COMPUTE THE LAPLACIAN, L.
1
1 −4 1
1
ACCUMULATE THE LAPLACIAN HISTOGRAM hist(L)
hist(L)
LOCATE THE MAXIMAL 2ND DERIVATIVE ON EACH SIDE OF hist(L)
d1, d2
t = f(d)
t1, t2

FIG. 4

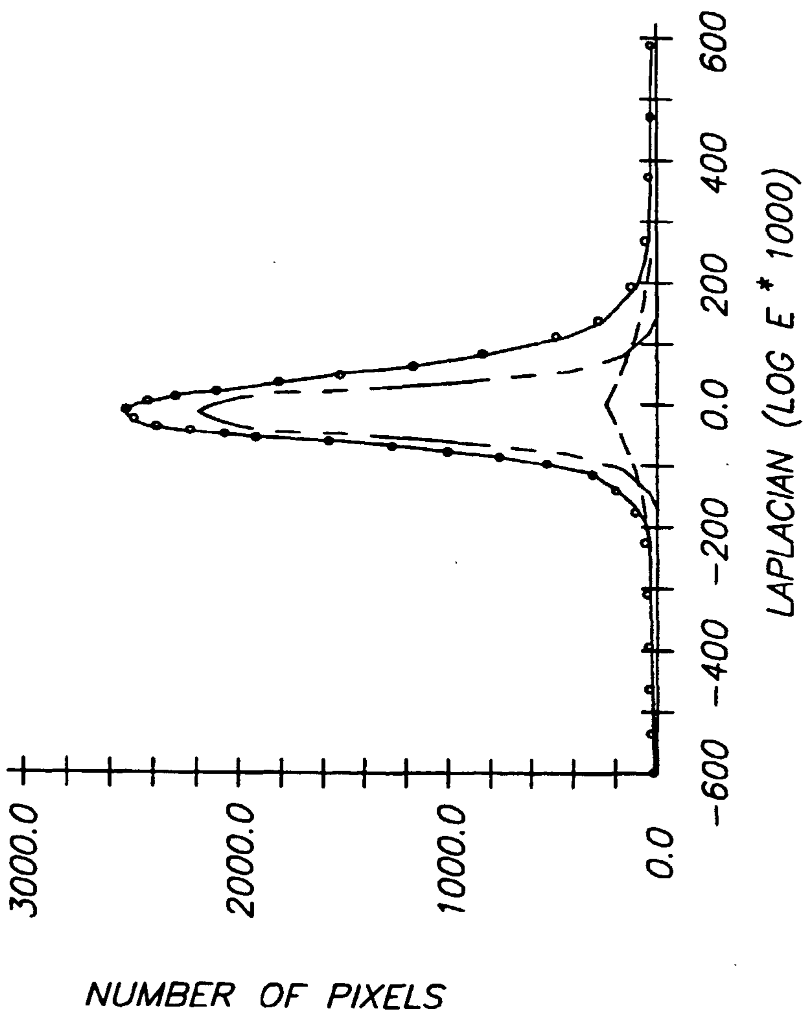
NUMBER OF PIXELS
3000.0
2000.0
1000.0
0.0
-600
-400
-200
0.0
200
400
600
LAPLACIAN (LOG E * 1000)

FIG. 5

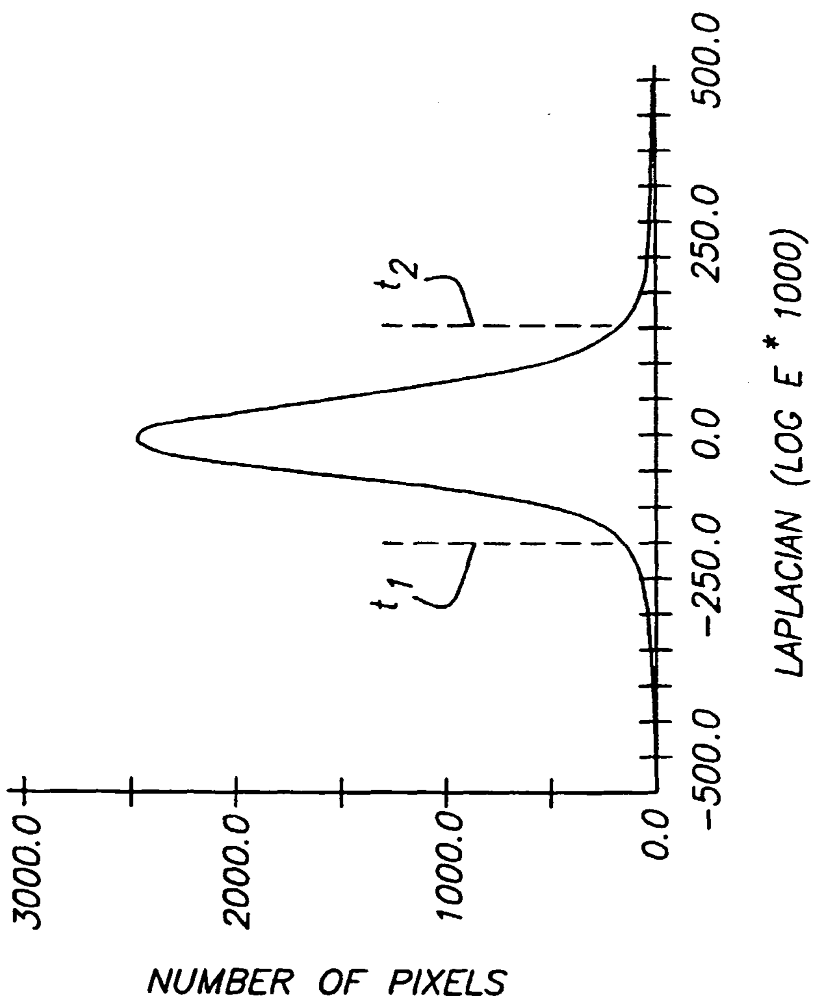
NUMBER OF PIXELS
3000.0
2000.0
1000.0
0.0
−500.0
−250.0
0.0
250.0
500.0
LAPLACIAN (LOG E * 1000)
t_1
t_2

FIG. 6

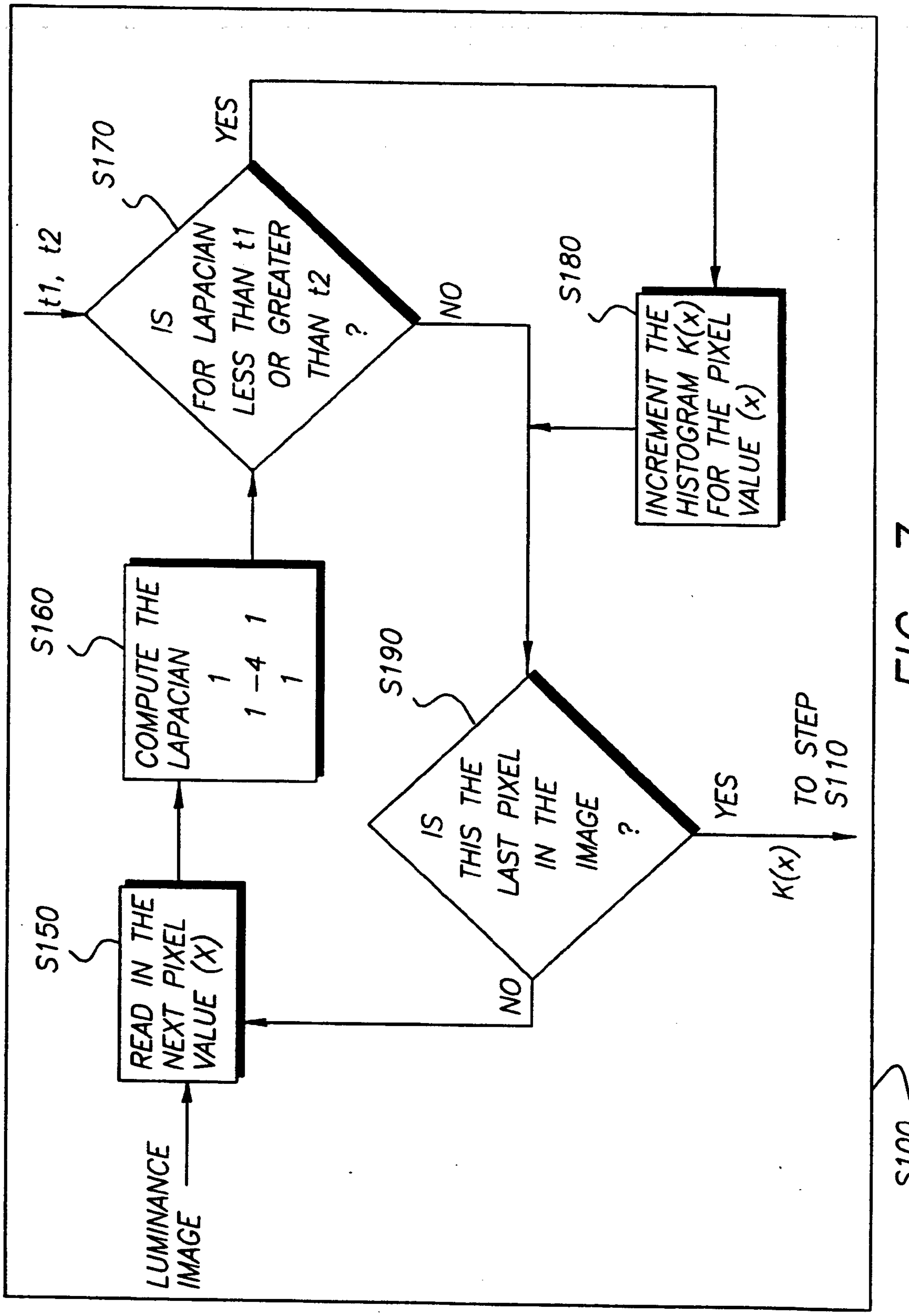
LUMINANCE IMAGE
S150
READ IN THE NEXT PIXEL VALUE (X)
S160
COMPUTE THE LAPACIAN
1
1 −4 1
1
t1, t2
S170
IS FOR LAPACIAN LESS THAN t1 OR GREATER THAN t2 ?
YES
NO
S180
INCREMENT THE HISTOGRAM K(x) FOR THE PIXEL VALUE (x)
S190
IS THIS THE LAST PIXEL IN THE IMAGE ?
YES
NO
K(x)
TO STEP S110
S100

FIG. 7

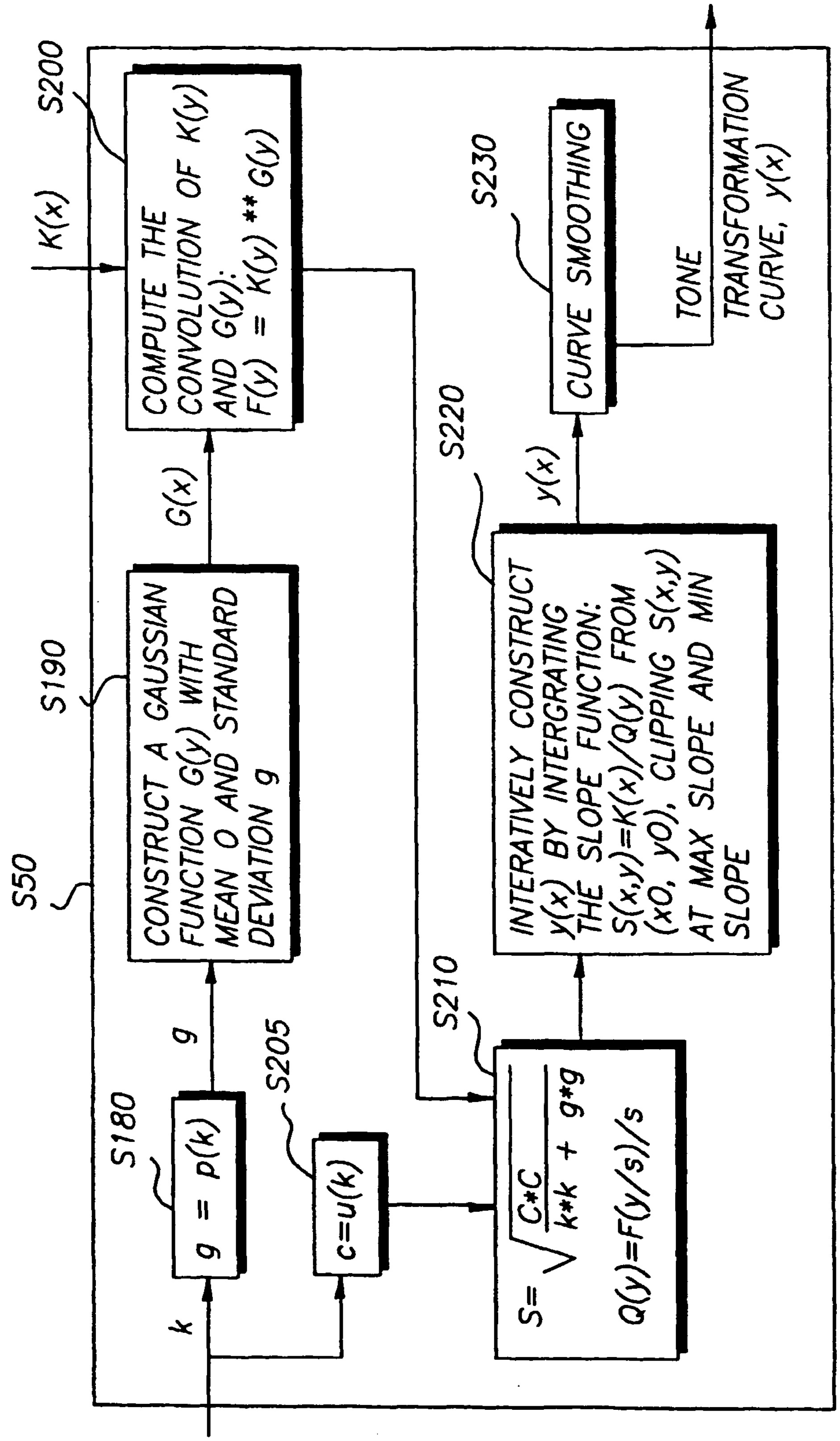
S50
S180
k
g = p(k)
g
S190
CONSTRUCT A GAUSSIAN FUNCTION G(y) WITH MEAN O AND STANDARD DEVIATION g
G(x)
S200
K(x)
COMPUTE THE CONVOLUTION OF K(y) AND G(y): F(y) = K(y) ** G(y)
S205
c=u(k)
S210
S= √(C*C / (k*k + g*g))
Q(y)=F(y/s)/s
S220
INTERATIVELY CONSTRUCT y(x) BY INTERGRATING THE SLOPE FUNCTION: S(x,y)=K(x)/Q(y) FROM (xO, yO), CLIPPING S(x,y) AT MAX SLOPE AND MIN SLOPE
y(x)
S230
CURVE SMOOTHING
TONE TRANSFORMATION CURVE, y(x)

FIG. 8

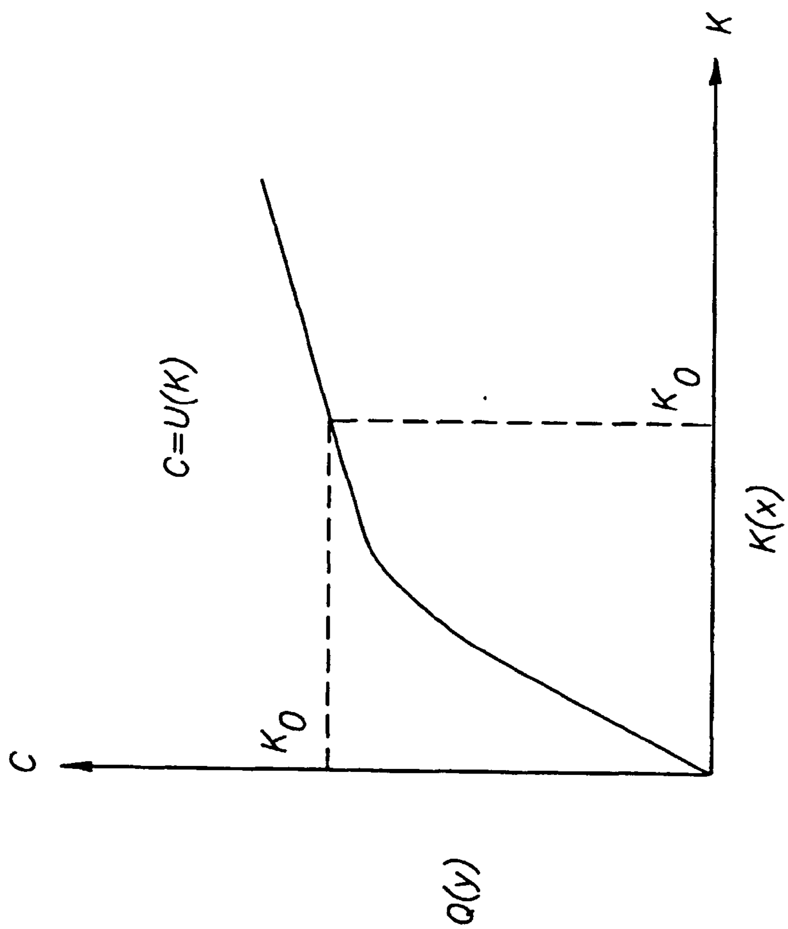

FIG. 9

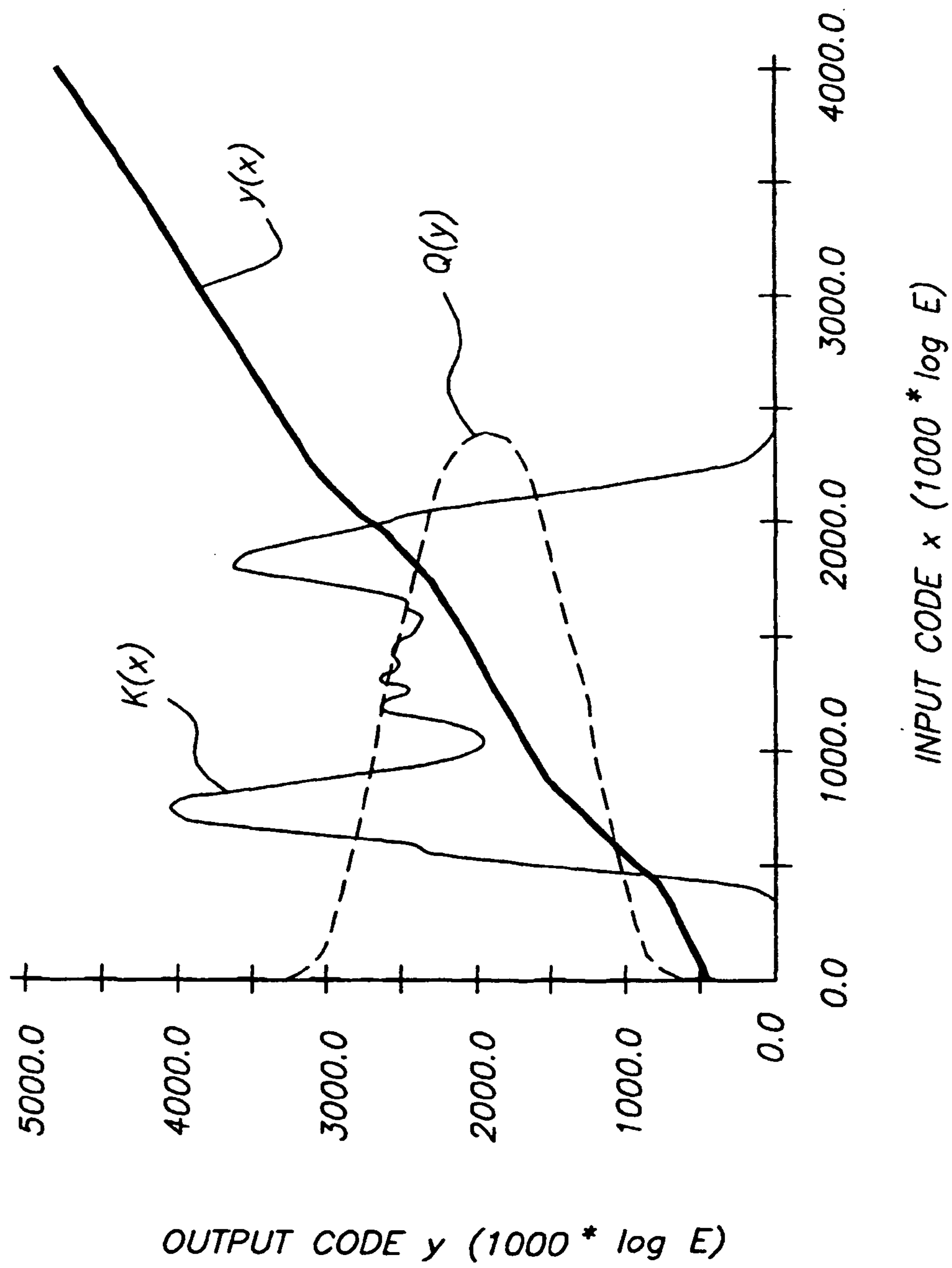
y(x)
Q(y)
K(x)
0.0
1000.0
2000.0
3000.0
4000.0
5000.0
INPUT CODE x (1000 * log E)
OUTPUT CODE y (1000 * log E)

FIG. 10

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 848 545 A3

(12) **EUROPEAN PATENT APPLICATION**

(88) Date of publication A3:
02.08.2000 Bulletin 2000/31

(43) Date of publication A2:
17.06.1998 Bulletin 1998/25

(21) Application number: **97203731.1**

(22) Date of filing: **28.11.1997**

(51) Int. Cl.^{7}: **H04N 1/407**, G06T 5/40

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **10.12.1996 US 763270**

(71) Applicant: **EASTMAN KODAK COMPANY**
Rochester, New York 14650 (US)

(72) Inventors:
- **Lee, Hsien-Che, c/o Eastman Kodak Company Rochester, New York 14650-2201 (US)**
- **Kwon, Heemin, c/o Eastman Kodak Company Rochester, New York 14650-2201 (US)**

(74) Representative:
Lewandowsky, Klaus, Dipl.-Ing.
Kodak Aktiengesellschaft,
Patentabteilung
70323 Stuttgart (DE)

(54) **Method for estimating and adjusting digital image contrast**

(57) A method of estimating the scene contrast from a digital image, the method comprises the steps of: forming a Laplacian histogram distribution; determining, from the Laplacian histogram, first and second thresholds which eliminate substantially uniform areas or a substantially textured portion of the digital image; selecting pixels which are based on the first and second thresholds from the digital image; forming a histogram from the sampled pixels; computing a standard deviation of the sampled histogram; and estimating contrast of the digital image by comparing the computed standard deviation with a predetermined contrast for determining contrast of the input image in relationship with the predetermined contrast.

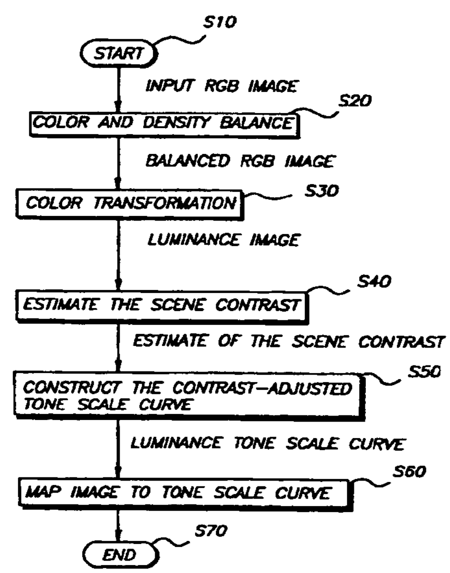

FIG. 2

EP 0 848 545 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 20 3731

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	US 5 544 258 A (LEVIEN RAPHAEL L) 6 August 1996 (1996-08-06) * column 3, lines 6-38; figure 8 * ---	1,5,8,9	H04N1/407 G06T5/40
D,A	US 4 731 671 A (ALKOFER JAMES S) 15 March 1988 (1988-03-15) * summary; column 2, lines 43-58; column 5, lines 12 - column 6, line 46; column 8, lines 23-40; figures 4,7,11 * ---	1,5,8,9	
A	FR 2 293 011 A (FRANCE ETAT) 25 June 1976 (1976-06-25) * page 4, lines 9-37 * ---	1,5,9	
A	US 5 426 517 A (SCHWARTZ MICHAEL) 20 June 1995 (1995-06-20) * summary; column 5, lines 18-66; column 7, lines 8-34 * ---	1,5,9	
A	US 4 764 971 A (SULLIVAN JAMES R) 16 August 1988 (1988-08-16) * summary; column 1, line 63 to column 2, line 15; column 9, lines 9-17 * ---	1,5,9	TECHNICAL FIELDS SEARCHED (Int.Cl.6) H04N
A	US 4 745 465 A (KWON HEEMIN) 17 May 1988 (1988-05-17) * summary; figures 1 and 2 * ---	1,5,9	
A	BLITON A C ET AL: "MICROSCOPIC MOTION ANALYSIS: LAPLACIAN-OF-GAUSSIAN MASKS FOR SUBPIXEL EDGE DETECTION" PROCEEDINGS OF THE ANNUAL INTERNATIONAL CONFERENCE OF THE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY,US,NEW YORK, IEEE, vol. CONF. 10, page 1098-1099 XP000571515 * summary; page 1098, whole right-hand column * --- -/--	1,5,9	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	26 May 2000	With, F

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 20 3731

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	DE 43 09 879 A (HELL AG LINOTYPE) 21 October 1993 (1993-10-21) * summary; page 4, line 44 to page 5, line 20 *	1,5,9	
A	VOSSEPOEL A M ET AL: "ADAPTIVE HISTOGRAM EQUALIZATION USING VARIABLE REGIONS" PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION. (ICPR),US,WASHINGTON, IEEE COMP. SOC. PRESS, vol. CONF. 9, 1988, pages 351-353, XP000093898 ISBN: 0-8186-0878-1 * page 351, left-hand column, third alinea of paragraph "1. Introduction" *	8	
A	DE 40 04 948 A (AGFA GEVAERT AG) 22 August 1991 (1991-08-22) * column 1, line 51 to column 2, line 9; column 3, line 65 to column 4, line 2; figures 2a, 2b and 2c *	8	TECHNICAL FIELDS SEARCHED (Int.Cl.6)
A	JAIN, Anil K.: "Fundamentals of Digital Image Processing", Prentice Hall, Englewood Cliffs, NJ, ISBN 0-13-336165-9, 1989, pages 233-244 * paragraph "Histogram Specification", pages 243 and 244 * XP002138831	8	

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	26 May 2000	With, F

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 20 3731

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	HOYER A ET AL: "DIGITAL IMAGE ENHANCEMENT" PHILIPS TECHNICAL REVIEW,NL,PHILIPS, EINDHOVEN, vol. 38, no. 11/12, 1978, pages 298-309, XP000865777 ISSN: 0031-7926 * paragraph "Contrast manipulation" from page 300 to page 304 * -----	8	
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	26 May 2000	With, F

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

European Patent Office

LACK OF UNITY OF INVENTION SHEET B

Application Number
EP 97 20 3731

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

1. Claims: 1-7, 9, 10

   method for excluding pixels of a digital image

2. Claim : 8

   finite difference method for constructing a transformation curve

European Patent Office

Application Number
EP 97 20 3731

CLAIMS INCURRING FEES

The present European patent application comprised at the time of filing more than ten claims.

☐ Only part of the claims have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims and for those claims for which claims fees have been paid, namely claim(s):

☐ No claims fees have been paid within the prescribed time limit. The present European search report has been drawn up for the first ten claims.

LACK OF UNITY OF INVENTION

The Search Division considers that the present European patent application does not comply with the requirements of unity of invention and relates to several inventions or groups of inventions, namely:

see sheet B

☒ All further search fees have been paid within the fixed time limit. The present European search report has been drawn up for all claims.

☐ As all searchable claims could be searched without effort justifying an additional fee, the Search Division did not invite payment of any additional fee.

☐ Only part of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the inventions in respect of which search fees have been paid, namely claims:

☐ None of the further search fees have been paid within the fixed time limit. The present European search report has been drawn up for those parts of the European patent application which relate to the invention first mentioned in the claims, namely claims:

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 97 20 3731

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

26-05-2000

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5544258 A	06-08-1996	NONE	
US 4731671 A	15-03-1988	DE 3672184 D EP 0222825 A JP 62500757 T WO 8606905 A	26-07-1990 27-05-1987 26-03-1987 20-11-1986
FR 2293011 A	25-06-1976	NONE	
US 5426517 A	20-06-1995	NONE	
US 4764971 A	16-08-1988	NONE	
US 4745465 A	17-05-1988	DE 3776655 A EP 0298978 A JP 1501909 T WO 8706089 A	19-03-1992 18-01-1989 29-06-1989 08-10-1987
DE 4309879 A	21-10-1993	NONE	
DE 4004948 A	22-08-1991	NONE	

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

THIS PAGE BLANK (USPTO)

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 02 00 0691

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
X	BRAUN G J ET AL: "Image lightness rescaling using sigmoidal contrast enhancement functions" JOURNAL OF ELECTRONIC IMAGING, vol. 8, no. 4, October 1999 (1999-10), pages 380-393, XP002303220	1-4, 8-14, 18-20	G06T5/40
Y	* pages 381-382; pages 390-391 *	5-7, 15-17	
Y	US 4 975 970 A (ZETTEL HUBERT A ET AL) 4 December 1990 (1990-12-04) * column 2, line 62 - column 3, line 30 *	5-7, 15-17	
A	EP 0 848 545 A (EASTMAN KODAK CO) 17 June 1998 (1998-06-17) * page 3, line 11 - line 25; figure 2 *	1-7, 11-17	
A	KOBAYASHI Y ET AL: "A high fidelity contrast improving model based on human vision mechanisms" MULTIMEDIA COMPUTING AND SYSTEMS, 1999. IEEE INTERNATIONAL CONFERENCE ON FLORENCE, ITALY 7-11 JUNE 1999, LOS ALAMITOS, CA, USA,IEEE COMPUT. SOC, US, 7 June 1999 (1999-06-07), pages 578-584, XP010519452 ISBN: 0-7695-0253-9 * the whole document *		

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

G06T
H04N

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
The Hague	29 November 2004	Reise, F

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

3

THIS PAGE BLANK (USPTO)